

## Case-Based Reasoning in Design: An Apologia

Kirt Pulaski

Martin Marietta Manned Space Systems  
Post Office Box 29304 D3691  
New Orleans, Louisiana 70189

Position 1: The process of generating solutions in problem solving is viewable as a design task.  
Position 2: Case-based reasoning is a strong method of problem solving.  
Position 3: A synergism exists between case-based reasoning and design problem solving.  
This paper presents and defends these three positions.

### 1 Introduction

#### *The Problem*

Design issues are omnipresent in everyone's day-to-day activities. People design business deals, they design sports strategies, they design physical objects, and they design an innumerable array of other tangible and intangible things. The issues which make designing difficult are basically two-fold: the design must serve some function, and it must do so while satisfying some set of constraints.

#### *The Paradigm*

Case-Based Reasoning (CBR) is a rather unique paradigm of artificial intelligence which weaves the history of experience into problem solving. Unlike other techniques which blindly and repeatedly solve new problems from scratch, CBR consults a memory of past cases to avoid re-solving recognizable problems. The quality and efficiency of reasoning gradually increases as new cases are acquired, analyzed, and added to the CBR memory.

#### *The Solution*

Experience is the most important factor which hones design skills, enabling the mastery of a core set of principles, rules, laws and techniques. Design experience is inseparable from design expertise, therefore it seems only natural that the next generation of intelligent computer-aided design tools should utilize CBR techniques.

#### *The Apologia*

The remaining sections of this paper build a case for case-based reasoning. The domain of discussion is CBR in engineering design. The goal is to get the research community involved in some issues and to get practitioners involved in some applications. By accelerating the inevitable (i.e., the use of CBR in design), beneficial applications for space technology can be realized sooner rather than later.

Section 2 describes a hypothetical session with an intelligent CBR design tool. The section whets the appetite and motivates the need for such a system. Section 3 details our current view of the engineering design process, without intelligent computers. Section 4 presents our proposed view of the same process, with intelligent computers.

Section 5 introduces the fetus of our research: the functional model of the CBR Designer's Assistant. Section 6 describes each module in more detail. Section 7 compares our model with other research projects and gives a compendium of work in the field. Section 8 concludes with some discussion and recommendations.

## **2 The Scenario**

To help set the stage, a typical interactive session between a designer and an intelligent computer will be presented. Let us call the system the CBR Designer's Assistant.

A design task is defined for a part belonging to a family of parts. The design requirements and constraints are presented to the designer, who comes to the design workstation and enters the CBR Designer's Assistant environment.

The designer requests assistance in performing a new design. A search is initiated in the CBR memory for previous cases of designs with the same or similar requirements and constraints. The system opens a new case, begins recording the design session, and prepares itself to track/guide the designer's goals throughout the session.

The designer selects a part from the list of retrieved cases and it is presented on the CAD screen. The designer browses the features, design decisions, and other knowledge associated with the past case. The designer selects and browses other similar designs (if there are any) until the closest match is found. If no relevant cases are found, the designer inputs features-based parametric commands to generate a new seed design.

The designer takes actions (makes decisions) within the system such as changing feature parameters, deleting features, adding features, editing relations, and so on. As this activity is performed, the system monitors and attempts to explain the actions (rationale) based upon its expectations. If needed, the system makes suggestions to the designer, guiding the design process.

The system activates a special design knowledge capture facility to prompt the designer to assist in the construction of explanations for actions which the system cannot self-explain. The facility associates the explanation information with the current case which will eventually be stored in memory. All aborted designs and dead ends are also captured and noted to enable the avoidance of failures in the future (through reminding).

The design evolves by specification/modification of features and parameters, which leads to more reminders and brings to bear pertinent lessons learned from the past. The embedded knowledge in the reminders enforces considerations related to form, functionality, production cost, materials, producibility, inspectability, and so on.

When the designer indicates to the system that the current preliminary design is satisfactory the resultant case is finalized and stored into the CBR memory.

The CBR Designer's Assistant is then prepared to digitally output/convert the information for the completed design case, at the designer's request, to downstream product definition systems. If the designer is not finished but wants to stop work, the system closes the design session and stores the unsatisfied goals with the case in memory. The designer can then resume the session at a later time and/or get progress updates.

### 3 Without Machine Intelligence

Figure 1 illustrates our functional model of the design process. In both Figure 1 and Figure 2 the heavy arrows represent the main input/output flow of producing a design. The light arrows represent the input/output growth of design experience. The lines without arrows represent mechanisms and controls that support the design process. The shaded regions represent the background enhancements that concurrent engineering concepts provide. The rounded boxes indicate non-computer activities, square boxes indicate computer-assisted activities.

The design process begins (cf. Figure 1) with the definition of the design task. This definition consists of requirements and constraints which are input to the designer. The designer combines creativity and innovation with an experiential design history to generate a new preliminary design. The design history is used to retrieve lessons learned and other relevant knowledge to increase the quality of the preliminary design.

Two mechanisms are then applied iteratively to evolve the design into a preliminary product definition. Problem solution tools are applied to specify and analyze solution concepts (e.g., geometric modeling, "what-if" parameter manipulation, rules and principles, and performance analyses). Design process management is applied to communicate decisions and reasoning (e.g., design plan, goal tracking, design reviews, and recording of design decision rationale).

The resulting preliminary product definition is then used to update the experiential design history. This ensures that future designs can benefit from the growth of design experience and lessons learned.

The process of design must blend many considerations and expertise from multiple disciplines (e.g., engineering, production, cost estimation, and quality). Many design groups choose to utilize a concurrent engineering approach, bringing scrutiny from other disciplines into the evolving preliminary design process. This concurrent approach greatly increases the efficiency of the design process. It saves time and money which is normally spent in implementing post-preliminary design changes (inherent in the non-concurrent approach).

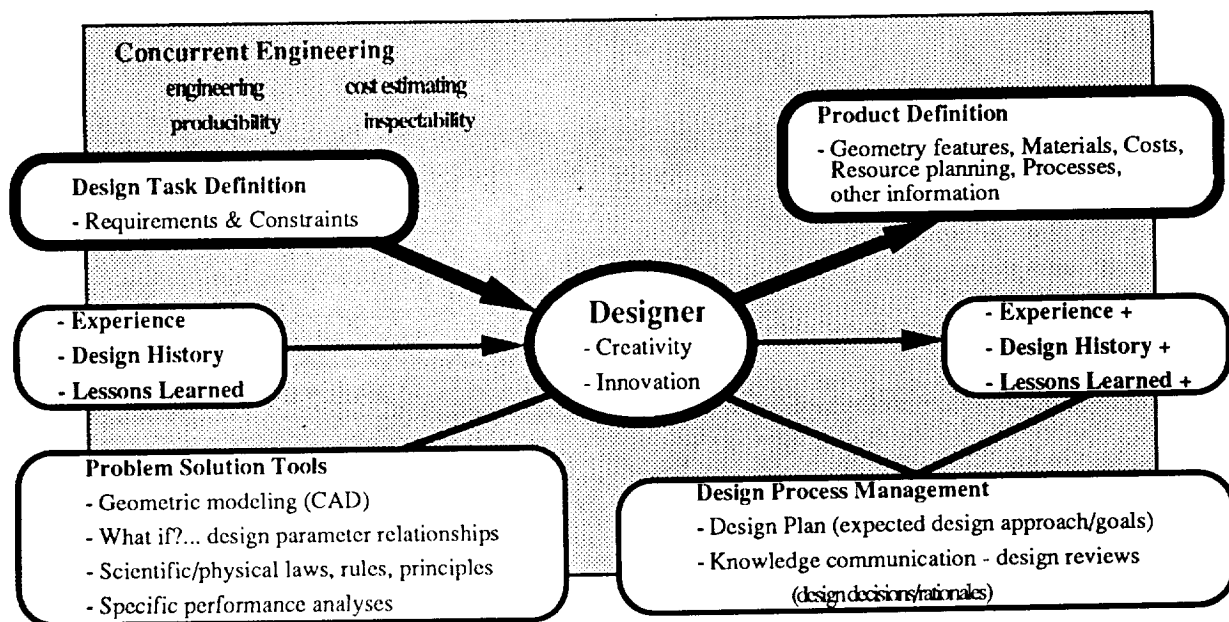


Figure 1: Without Machine Intelligence

#### 4 With Machine Intelligence

Figure 2 shows an improved view of the Figure 1 model where some parts of the process are computer-assisted. The position of this paper is to promote the inclusion of CBR and other machine intelligence techniques when implementing this model.

The new model maps the design process depicted in Figure 1 onto a computer architecture as shown in Figure 2. The hashed line surrounding the square boxes defines the boundaries of the CBR Designer's Assistant. Creativity and innovation is still left to the designer, only now the computer allows the designer to focus more on these and less on the other mechanisms.

Figure 2 shows that the growing experiential design history is implemented as a CBR Module. This module also contains concurrent engineering knowledge. The problem solution tools are implemented as an Intelligent Computer-Aided Design (IntCAD) Module, also containing concurrent engineering knowledge. The design process management is implemented as a Design Session Manager (DSM) Module.

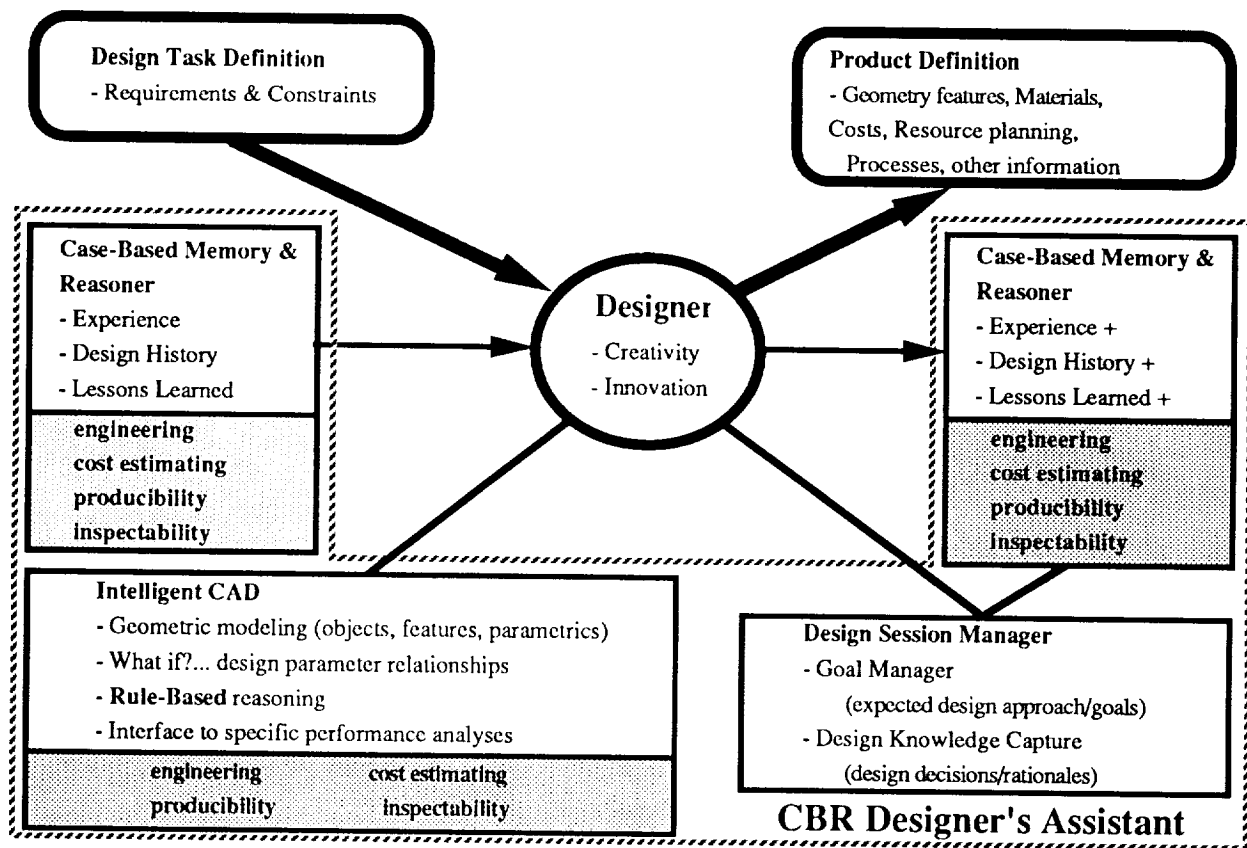
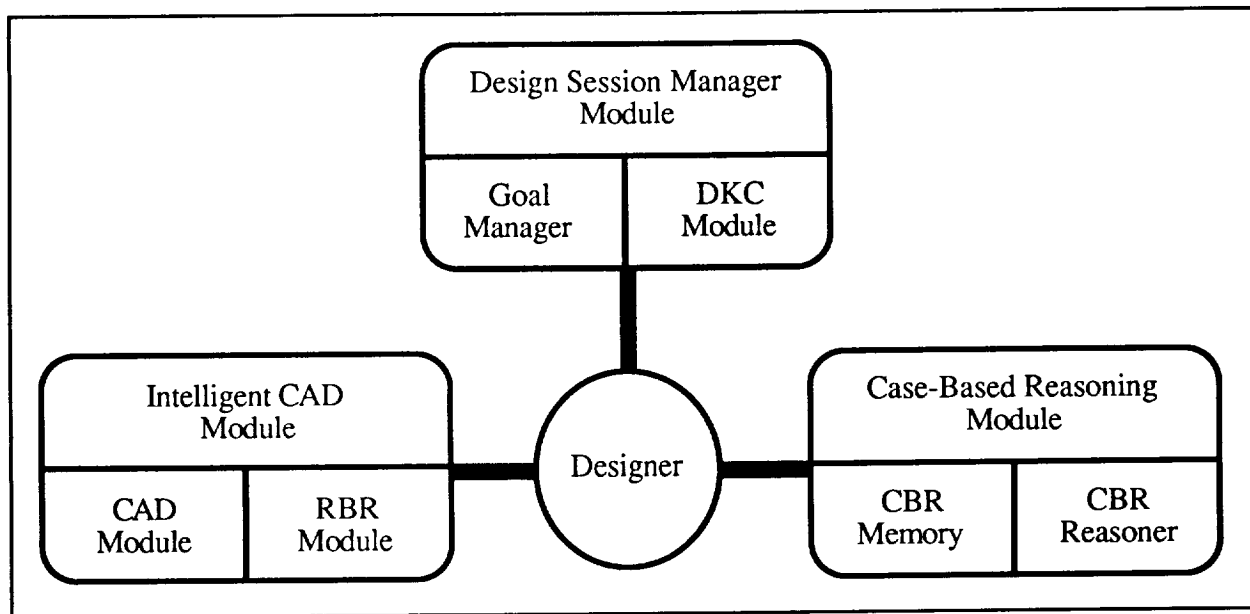


Figure 2: With Machine Intelligence



**Figure 3:** *Case-Based Reasoning Designer's Assistant*

## 5 The CBR Designer's Assistant

Figure 3 shows the architecture of the CBR Designer's Assistant. This section introduces the machine intelligence techniques which are used to provide the required functionality of each module.

The CBR Module has a memory component containing the design history, lessons learned, and concurrent engineering experience and knowledge. It also has a reasoning component to analyze relevant cases from memory and make pertinent suggestions to the designer. The memory grows and learns over time as the system is used. The machine intelligence techniques to develop this module come from the paradigm of case-based reasoning.

The IntCAD Module has a features-based parametric CAD component to help the designer generate preliminary designs. It also has a rule-based reasoner which enables "what-if" parametric analyses, interfaces to performance analysis routines, and manipulates concurrent engineering knowledge. The machine intelligence techniques to develop this module are rule-based reasoning and object-oriented programming.

The DSM Module has a goal manager to provide the design plan enforcement mechanism. It also has a design knowledge capture component to record decision rationale. The machine intelligence techniques used in this module are expectation-based processing and explanation-based processing.

## 6 The Modules

### *CBR Module*

CBR is a paradigm of problem solving which uses past solutions and lessons learned to solve new problems. Researchers agree that the quality of decision-making particular to a design

can be vastly improved if features from similar past design cases are considered. In fact, the training of novice designers is case-based [Sycara 89c].

Our experience (including analysis of other researchers' results) indicates that CBR offers the following general advantages:

- 1) CBR is an efficient "jump-start" technique: focused problem solving is achieved faster, through reminders, than blind search (such as rule-based) techniques.
- 2) Reminded cases provide efficient "repair" strategies [Sycara 89b] when debugging partial solutions: previous successes, failures, and lessons learned are applicable piece-wise, even when the reminded pieces are from different (but related) domains.
- 3) Cases enable learning: feedback with respect to the success or failure of a proposed solution, along with justifications as to why, are recordable and become new cases in memory to benefit future problem solving.

Simply stated, advantages 1 and 3 provide intelligent "book ends" to a problem solving session; advantage 2 steers the process in between. The following paragraphs in this section present how these general advantages apply in the CBR Designer's Assistant.

Advantage 1 is used to "jump-start" the design process. A features-based definition of a part to be designed is input. The most relevant design matching those features is retrieved from memory. Presenting the retrieved design to the user/designer provides an immediate seed preliminary design to focus the design process.

Advantage 2 is used to optimize the preliminary design piece-wise. As the user/designer modifies pieces of the preliminary design (assuming it did not exactly match the input features), additional similar pieces are retrieved from memory. The reminded pieces help repair the desired changes by reporting violated constraints (such as producibility, cost, or inspectability) from the past and adapting their solutions to the present.

Advantage 3 enables the CBR Designer's Assistant to learn. Explanations and justifications as to how/why design decisions are made (both good and bad) are recorded to benefit future similar designs.

The introduction to the DARPA Machine Learning Program Plan [DARPA 89] lists five advantages of CBR: performance enhancement, uncomplicated learning, cases serve as generalizations, scalability of methods, and easier knowledge acquisition. We agree with these, however space here does not permit a detailed comparison.

### *IntCAD Module*

Currently there exist software tools which enable features-based parametric design. Some of the more novel tools incorporate object-oriented programming and rule-based knowledge as well (ICAD from ICAD, Inc., and Wisdom Ware from Wisdom Systems, Inc.).

There are two basic issues with respect to the IntCAD Module. The first issue relates to the difficulty of customizing these off-the-shelf products. Considerable effort, expense, and commitment by engineering groups is required, however the results are well worth it.

The other issue is an integration concern. After customization, the tool must integrate with the rest of the modules in the CBR Designer's Assistant.

## *DSM Module*

The DSM Module interfaces with the designer and manages the case-based and rule-based reasoning processes. Within the DSM Module are the Goal Manager and the Design Knowledge Capture (DKC) Module.

A popular theory of human intelligence states that learning involves several steps:

- 1) Expectation      First there needs to be some form of expectation.
- 2) Surprise        When things do not happen as expected, surprise occurs.
- 3) Explanation     A surprise spawns the desire for an explanation.
- 4) Learning        The analysis of an explanation leads to learning.

Step 1 is known as expectation-based processing. Representations of goals enable the Goal Manager to build expectations of what the user/designer will do.

Step 2 occurs when an expectation is not met. The ways in which the satisfied and unsatisfied goals relate to each other are used by the Goal Manager to further identify the type of surprise.

Step 3 is known as explanation-based processing. Representations of rationale enable the DKC Module to prompt the user/designer for acceptable explanations.

Step 4 occurs as a result of building an acceptable explanation for a surprise. The DKC Module records the explanation and associates it with the current design case. The case gets stored in the memory thereby learning an explanation which can be used for future cases.

## **7 State-of-the-Research**

There are two central issues in CBR research: retrieving relevant cases from memory, and reasoning from the cases which are retrieved. For comparison, [DARPA 89] details three central issues: index selection, rank-ordering reminders, and adaptation. We would include the first two of these within our retrieval issue.

The most relevant research that we are aware of is that which is being performed by Sycara and Navinchandra at the Robotics Institute of Carnegie Mellon University. Their research is aimed at integrating case-based reasoning and qualitative engineering design. The domain of their research is in the area of the automated design of mechanical assemblies.

Another related research effort is being performed by Finger et al. [Finger 88]. Their research effort, named The Design Fusion Project, is large-scale in which they address automated design from a product life-cycle view. The domain of their research is electro-mechanical assemblies.

Martin Marietta Laboratories is working on the larger issues of how to manage and integrate concurrent engineering on a broad spectrum of design types, domains, activities and users in a project called Integrated Concurrent Engineering (ICE) [Mills 89]. The work is also addressing the issue of how to manage and integrate the large variety of design aids emerging from research laboratories including algorithmic, expert system-based, finite element-based, simulation-based, and so on. The domain of their research is mechanical assemblies.

There are numerous other intelligent design research efforts [Brown 85], [Bulkeley 89], [Dixon 88], [Hayes-Roth 83], [Ullman 88]. One last project in engineering design will be cited. The KIDS (Knowledge Integrated Design System) project at Wright-Patterson Aeronautical Laboratories [KIDS 89] is focused on many of the same issues as the ICE project. The domain of that research is process-oriented: rolling, forging, casting, and some advanced composites processes.

The domain of design is actually present in many human-based activities, not just engineering. Our research is related to many other research efforts which look at design from a non-engineering view: for example, two parties designing a mediation to a dispute [Simpson 85] or a chef designing a meal [Hinrichs 89], [Hammond 86]. The reader is referred to [DARPA 88] and [DARPA 89] which offer excellent compilations of work in CBR.

There are some researchers in the CBR community that have just recently begun to address the design domain. The sheer growth in the number of these relevant research projects, compared to just a year ago, indicates that CBR has the potential to offer many benefits to the domain of design. Synopses of a representative sample of these research projects follows.

Birnbaum and Collins [Birnbaum 89] consider "design themes" as part of an indexing vocabulary to provide cross-contextual reminders.

Goel and Chandrasekaran [Goel 89] work in the adaptation of previous design cases by considering the designer's functional model of his/her causal understanding of the behavioral and structural aspects of a design (called a device model).

Alexander et al. [Alexander 89] are formalizing a representation scheme (called a design tree) which allows design cases to be transformed into new designs using a calculus they have developed.

Barletta and Hennessy [Barletta 89] have developed a method which adapts pieces of previous cases to optimize the placement of composite parts to be cured in an autoclave.

Sycara and Navinchandra [Sycara 89a], [Sycara 89c] are using a multi-layered representation scheme (structural, functional, causal, and qualitative) to effect index transformation during reminding.

Our own work [Pulaski 88a], [Pulaski 88b], [Hightower 89] has been focused on building CBR memories automatically from a case base (a database plus a knowledge base), and then optimizing the memory using neural network processing.

Although outside the scope of this paper, it should be mentioned that the IntCAD Module and DSM Module each involve many issues and relate to other ongoing research. Perhaps the area most needy of a breakthrough is in design knowledge capture. We agree with Wechsler [Wechsler 86] that part of the solution needs to be explanation-based. The reader is referred to Freeman [Freeman 88] for an excellent overview of the design knowledge capture problem. Also, the DSM Module would benefit from a more sophisticated user model, beyond simply goal tracking.



## 8 Discussion

Several observations are notable regarding current work in CBR and integrated design. The ICE, KIDS, and Design Fusion projects are large-scale. Much of this and other work has focused on examples involving complex electro-mechanical assemblies. Although these projects should some day produce very significant results, the size of the efforts and the complexity of the domain will make progress very difficult.

We believe that there is a shortage of small-scale projects focused on simple designs. The study of CBR in design will benefit sooner by addressing the issues with simpler examples. It is our recommendation that single-component designs be considered, such as molds, fasteners, or composite parts. These examples will reduce the amount of complexities arising from sub-component interactions, yet results will still be beneficial and usable.

Another observation is that much work is geared toward a high level of design automation. Our position is that using machine intelligence to assist a designer (rather than replace him/her) will produce real benefits faster and at less expense than attempting to automate the construction of an original design (without including a human designer).

It is generally agreed that while spending only 5% of manufacturing costs during design, decisions made during design commit 95% of manufacturing costs. There are many stages throughout the design process which can benefit from machine intelligence, however we feel that the stage of preliminary design can benefit the most. Using CBR to bring downstream considerations into the preliminary design process will assist the designer to optimize the upstream design for cost as well as other considerations (e.g. quality, producibility, maintainability, repairability, and other life-cycle concerns).

## Acknowledgement

The author would like to extend his warmest thanks to Mr. Thomas J. Leahey who selflessly made himself available for numerous brainstorming sessions and discussions. Tom's expertise in the area of design contributed significantly to the ideas presented in this paper.

## References/Bibliography

- [Alexander 89] Perry Alexander, Gary Minden, Costas Tsatsoulis and J. Holtzman, "Storing Design Knowledge in Cases," in [DARPA 89], pp. 188-192.
- [Barletta 89] Ralph Barletta and Dan Hennessy, "Case Adaptation in Autoclave Layout Design," in [DARPA 89], pp. 203-207.
- [Birnbaum 89] Lawrence Birnbaum and Gregg Collings, "Reminders and Engineering Design Themes: A Case Study in Indexing Vocabulary," in [DARPA 89], pp. 47-51.
- [Brown 85] D. C. Brown and B. Chandrasekaran, "Expert Systems for a Class of Mechanical Design Activity," *Knowledge Engineering in Computer-Aided Design* (underlined), J. S. Gero (Editor), Elsevier Science Publishers B. V. (North-Holland), 1985, pp. 259-283.
- [Bulkeley 89] Debra Bulkeley, "Get Smart: Software that Knows the Answers," *Design News* (italics), February 27, 1989, pp. 72-80.
- [DARPA 89] Proceedings of the Case-Based Reasoning Workshop, Sponsored by DARPA, Pensacola Beach, Florida, May 31 - June 2, 1989, Morgan Kaufmann Publishers, San Mateo, California.

- [Dixon 88] J. R. Dixon, "Designing with Features: Building Manufacturing Knowledge into more Intelligent CAD Systems," in Proceedings of Manufacturing International '88, Atlanta, Georgia, April 17-20, 1988, pp. 51-57.
- [Finger 88] Susan Finger, Mark S. Fox, Dundee Navinchandra, Friedrich B. Prinz, and James R. Rinderle, "The Design Fusion Project: A Product Life-Cycle View for Engineering Designs," a proposal, May 26, 1988.
- [Freeman 88] Michael S. Freeman, "The Elements of Design Knowledge Capture," in Proceedings of AISA 88, November 15-16, Huntsville, AL, NASA Conference Publication 3013, pp. 39-46.
- [Goel 89] Ashock Goel and B. Chandrasekaran, "Use of Device Models in Adaptation of Design Cases," in [DARPA 89], pp. 100-109.
- [Hammond 86] Kristian Hammond, "Case-Based Planning: An Integrated Theory of Planning, Learning and Memory," Ph.D. Thesis, Yale University, New Haven, CT.
- [Hayes-Roth 83] Frederick Hayes-Roth, "Using Proofs and refutations to Learn from Experience," Machine Learning (underlined), Tioga Publishing Company, Palo Alto, California, 1983, pp. 221-240.
- [Hightower 89] Ron R. Hightower and Kirt Pulaski, "An Investigation of Enhancing ELMO with Neural Network Processing," poster session at DARPA CBR Workshop, Pensacola Beach, Florida, June 1, 1989.
- [Hinrichs 89] Thomas R. Hinrichs, "Strategies for Adaptation and Recovery in a Design Problem Solver," in [DARPA 89], pp. 115-118.
- [Kids 89] "KIDS: Knowledge Integrated Design System," RFP No. F33615-89-R-5619, Headquarters Aeronautical Systems Division, Wright-Patterson Air Force Base, Ohio, May 3, 1989.
- [Mills 89] John Mills, "ICE: Integrated Concurrent Engineering Project Overview," Martin Marietta Laboratories, Baltimore, Maryland, 1989 (unpublished).
- [Pulaski 88a] Kirt Pulaski, "ELMO: An Episodic Long-term Memory Organizer for Case-Based Reasoning," AAAI Case-Based Reasoning Workshop, St. Paul, Minnesota, August 23, 1988, pp. 107-112.
- [Pulaski 88b] Kirt Pulaski and Cyprian E. Casadaban, "Case-Based Reasoning: The Marriage of Knowledge Base and Data Base," in Proceedings of Fourth Conference on Artificial Intelligence for Space Applications, Huntsville, Alabama, November 15-16, 1988, NASA Conference Publication 3013, pp. 183-190.
- [Simpson 85] Robert L. Simpson, "A Computer Model of Case-Based Reasoning in Problem Solving: An Investigation in the Domain of Dispute Mediation," Ph. D. dissertation, School of Information and Computer Science, Georgia Institute of Technology, Atlanta, Georgia, Technical Report GIT-ICS-85/18, 1985.
- [Sycara 89a] Katia Sycara and Dundee Navinchandra, "Index Transformation and Generation for Case Retrieval," in [DARPA 89], pp. 324-328.
- [Sycara 89b] Katia Sycara and Dundee Navinchandra, "Representing and Indexing Design Cases," in Proceedings of IEA/AIE-89, Tullahoma, Tennessee, June 6-9, 1989.
- [Sycara 89c] Katia Sycara and Dundee Navinchandra, "Integrating Case-Based Reasoning and Qualitative Reasoning in Engineering Design," in Proceedings of AI in Engineering 89 Conference, Cambridge, U.K., July, 1989.
- [Ullman 88] David G. Ullman, Thomas G. Dietterich and Larry A. Stauffer, "A Model of the Mechanical Design Process Based on Empirical Data," *AI EDAM* (italics), 2(1), pp. 33-52, Academic Press Limited, 1988.
- [Wechsler 86] D. B. Wechsler, "An Approach to Design Knowledge Capture for the Space Station," in Proceedings of Space Station Automation II, sponsored by SPIE, Vol. 729, 1976, pp. 106-113.